

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

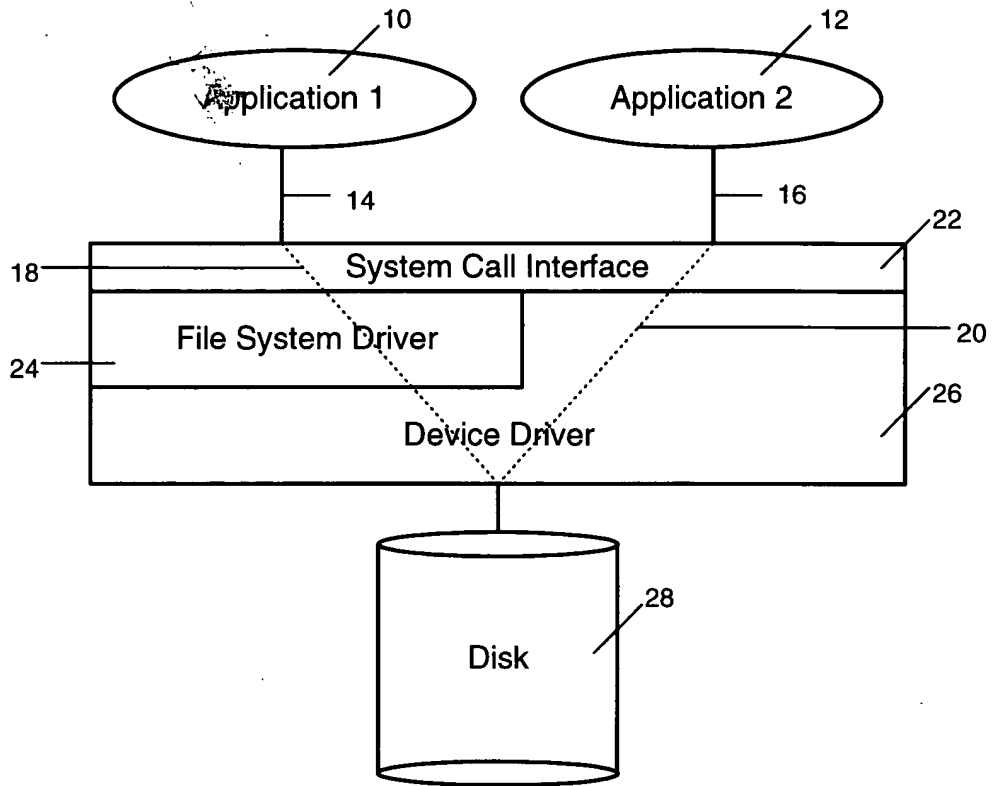


Figure 1.

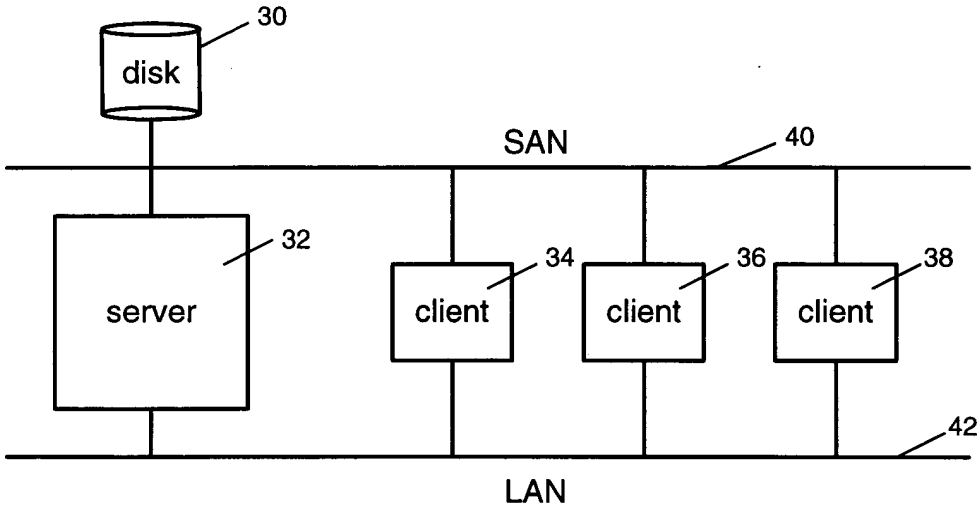


Figure 2.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

Event

Client

Server

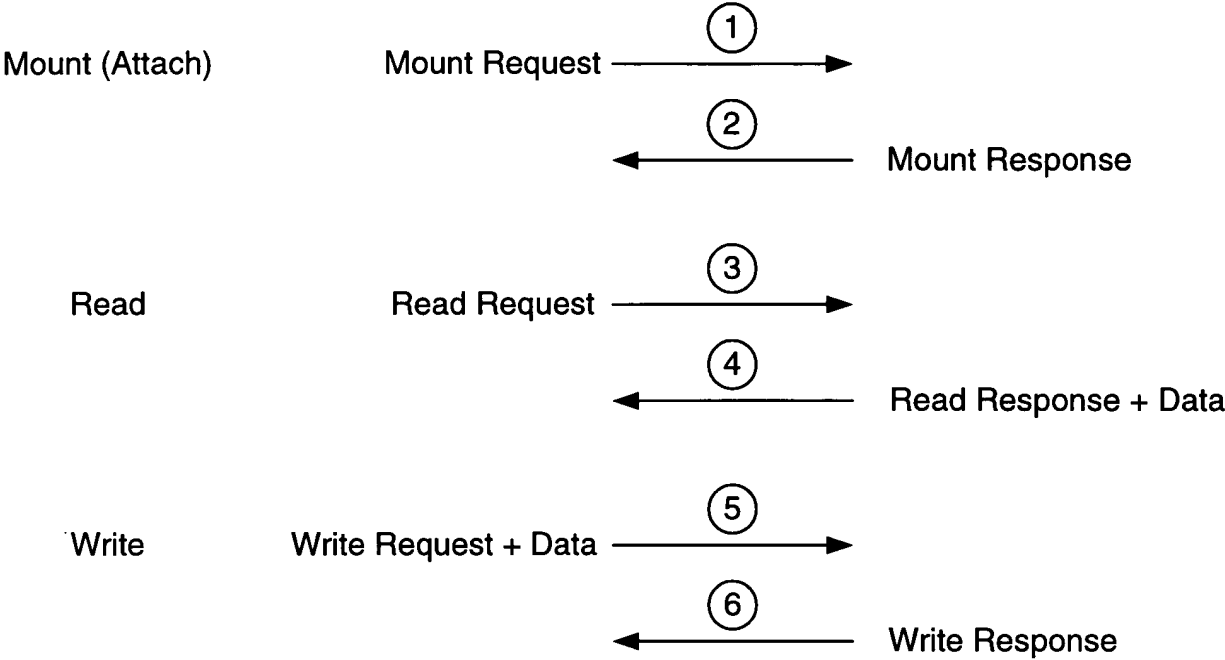


Figure 3.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

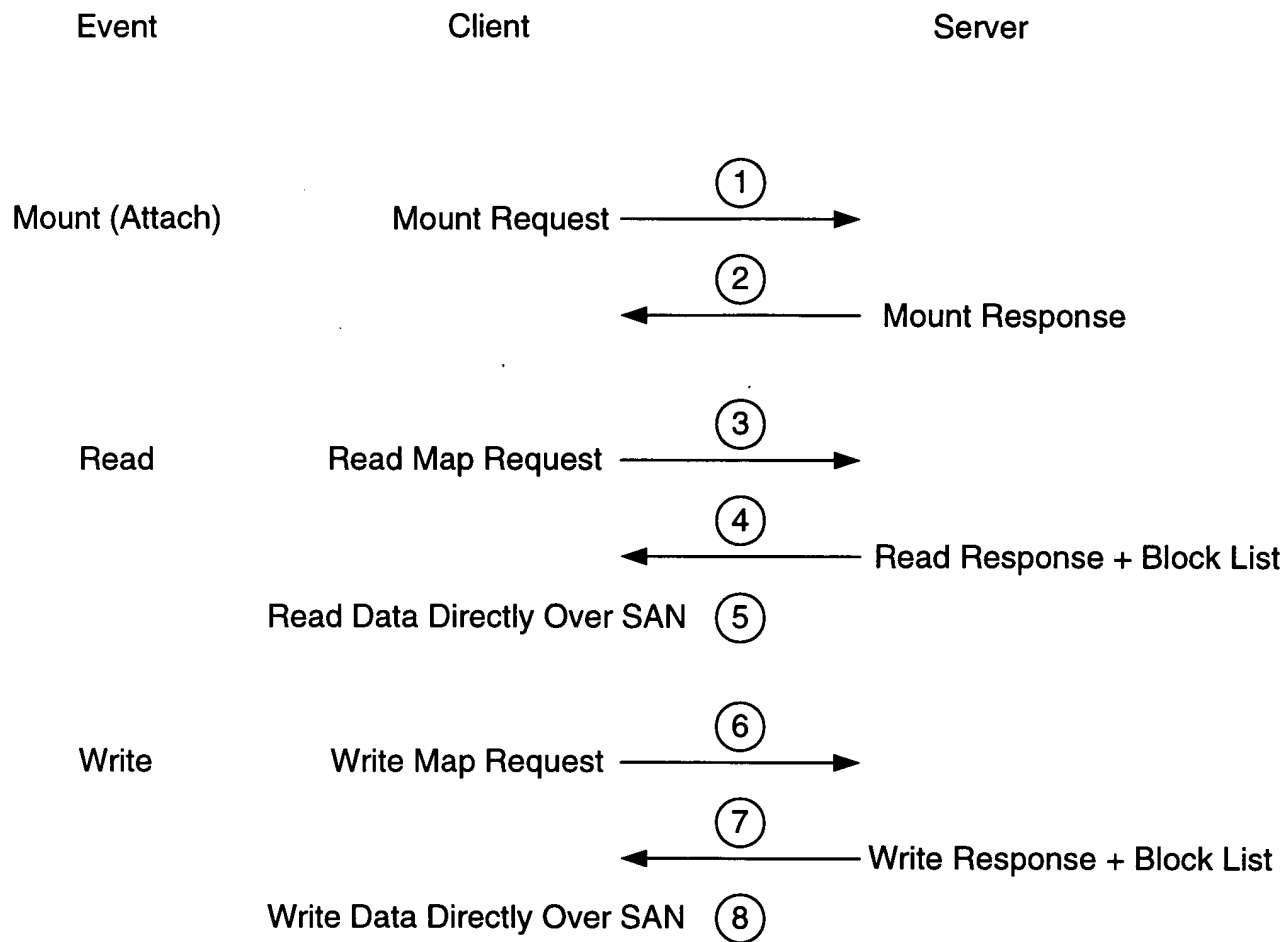


Figure 4.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

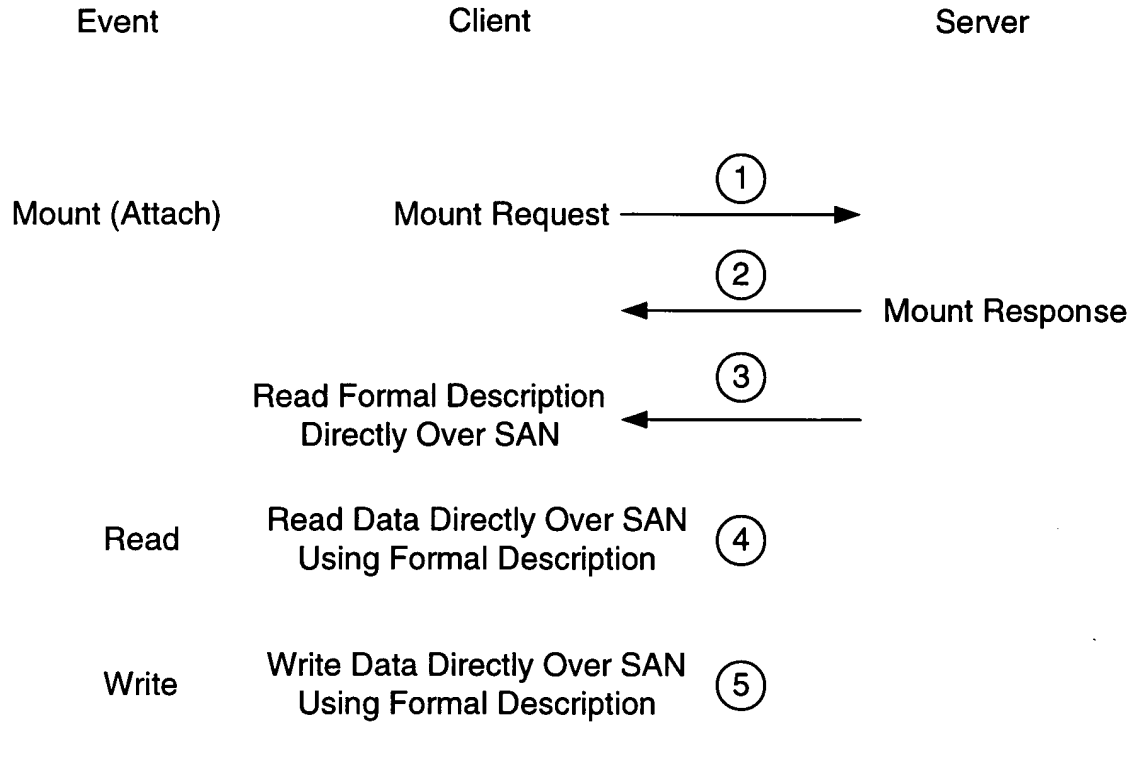


Figure 5.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

Feature	Network File System	SNIA Proposal	Self-Describing File System	Clustered File System
Serialization	server	server	server	clients
Disk Block Mapping	server	server	clients	clients
Block Allocation	server	server	server	clients
Disk Access	server	clients	clients	clients

Figure 6.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

```

<FS Parameters>
  <CONST NAME="BSIZE"> 1024 </CONST>
  <CONST NAME="INOSZ"> 64 </CONST>
  <CONST NAME="INOPB"> BSIZE / INOSZ </CONST>
  <CONST NAME="ISIZE"> {derived from super block} </CONST>
  <CONST NAME="STARTI"> 2 </CONST>
  <CONST NAME="PSTART"> {physical start of file system} </CONST>
  <CONST NAME="NADDR"> 13 </CONST>
  <CONST NAME="NDADDR"> 10 </CONST>
  <CONST NAME="NIADDR"> 3 </CONST>
  <CONST NAME="IADDRSZ"> 3 </CONST>
  <CONST NAME="IADDROFF"> 12 </CONST>
  <CONST NAME="ISIZEOFF"> 8 </CONST>
  <CONST NAME="BYTEORDER"> 0 </CONST>
  <CONST NAME="NSHIFT"> 8 </CONST>
  <CONST NAME="NINDIR"> BSIZE / 4 </CONST>
  <CONST NAME="NBPSCTR"> 512 </CONST>
</FS Parameters>

<MACRO NAME="LTOPBLK" PARAMS="BN"> BN * (BSIZE / NBPSCTR) </MACRO>

<FUNC NAME="iread">
  <BODY>
    int32
    iread(int16 ino, char *buf)
    {
      int32 bn;
      int32 boff;

      bn = (ino + (2 * INOPB - 1)) / INOPB;
      boff = (ino + (2 * INOPB - 1)) & (INOPB - 1);
      PREAD(LTOPBLK(bn), buf, BSIZE);
      return boff;
    }
  </BODY>
</FUNC>

<FUNC NAME="bmap">
  <BODY>
    int32
    bmap(char *ibuf, int32 off, int32 len, int32 *dbuf)
    {
      int32 sh;
      int32 i;
      int32 j;
      int32 bn;
      int32 blim;
      int32 nblk;
      int32 *bnp;
      int32 daddr[NADDR];
      char ib[BSIZE];
      char *cp;
      int32 naddr = 0;

      nblk = len + (BSIZE - 1) / BSIZE;
      if (nblk == 0)
        return 0;
    }
  </BODY>
</FUNC>

```

Figure 7.

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

```

/*
 * build an address array, converting from 3-byte
 * addresses to 4-byte addresses.
 */
cp = ibuf + IADDROFF;
for (i = 0, j = 0; i < IADDRSZ; i += IADDRSZ, j++) {
    if (BYTEORDER == 0)
        daddr[j] = cp[i]<<16|cp[i+1]<<8|cp[i+2];
    else
        daddr[j] = cp[i+2]<<16|cp[i+1]<<8|cp[i];
}

/*
 * Fill the dbuf array with the list of block numbers.
 */
while (len > 0) {
    bn = off / BSIZE;
    len -= BSIZE;
    off += BSIZE;
    if (bn < NADDR - NIADDR) {
        dbuf[naddr++] = daddr[bn];
        continue;
    }
    bn -= NDADDR;
    sh = 0;
    blim = 1;
    for (j = NIADDR; j > 0; j--) {
        sh += NSHIFT;
        blim <= NSHIFT;
        if (bn < blim)
            break;
    }
    if (j == 0)
        return naddr;
    ibn = daddr[NADDR-j];
    if (ibn == 0) {
        dbuf[naddr++] = 0;
        continue;
    }
    for (; j <= 3; j++) {
        sh -= NSHIFT;
        PREAD(LTOPBLK(ibn), ib, BSIZE);
        bnp = (int32 *)ib;
        i = (bn >> sh) & (NINDIR - 1);
        if (bnp[i] == 0)
            break;
        ibn = bnp[i];
    }
    dbuf[naddr++] = bnp[i];
}
return naddr;
}

```

</BODY>
 </FUNC>

Figure 8.